Microdrones VSM User Guide

UgCS 2.5.242

Copyright © 2015, Smart Projects Holdings Ltd

Contents

1 Microdrones VSM User Guide **1**

1.1 First time vehicle connection . . . 1

1.2 Mission execution specifics . . . 2

1.3 Command execution specifics . . . 2

1.4 Telemetry information specifics . . . 3

1.5 Fail-safe actions . . . 3

1.6 Configuration file . . . 3

1.6.1 Common parameters . . . 3

1.6.2 Serial port configuration . . . 4

1.6.3 Model name override . . . 4

1.7 Common configuration file parameters . . . 4

1.7.1 UgCS server configuration . . . 4

1.7.2 Logging configuration . . . 5

1.7.3 Serial port configuration . . . 5

1.7.4 Network connection configuration . . . 6

1.7.5 Mission dump path . . . 7

2 Disclaimer **7**

Copyright © 2015, Smart Projects Holdings Ltd

1 Microdrones VSM User Guide

1.1 First time vehicle connection

See Disclaimer.

Please follow these steps to connect an Microdrones vehicle to the U[g]CS:

1. Microdrones vehicle must be properly configured, calibrated and tested using tools and instruction from the official `Microdrones web site` prior to using it with U[g]CS. U[g]CS does not support initial configuration, setup and calibration of Microdrones vehicles.

2. Turn on the vehicle. There are two communication channels in a drone - uplink and downlink. Uplink channel is connected via USB-serial cable, plug it in the drone connector (you should connect to FC, not NC - connector flat side to the battery, see manufacturer manual for detailed explanation) and to the computer where VSM is running. The uplink channel can be used for mission uploading and commands execution. For telemetry reception you need downlink channel which is connected to the downlink wire on the drone bottom. You can use manufacturer provided downlink equipment which uses video transmitter sound channel for telemetry data transferring and downlink decoder ground unit. Alternatively you can connect your own USB-serial cable or radio-modem to the downlink wire. Before that you should switch the telemetry output to digital mode using a jumper on the drone board, see manufacturer manual for the details.

   For initial vehicle set up in U[g]CS you can connect either uplink or downlink whichever is more convenient for you.

   For all connection options you will have USB-serial device on the PC side so proper OS driver for virtual serial port should be installed. Please refer to your communication equipment manufacturer documentation about driver installation instructions.

3. Open *Vehicles* window in U[g]CS Client and wait until new vehicle appears there automatically. Either Uplink or Downlink connection should be available. Press *Gain control* and *Edit* to select corresponding vehicle profile and change the default vehicle name to be convenient for you:

Copyright © 2015, Smart Projects Holdings Ltd

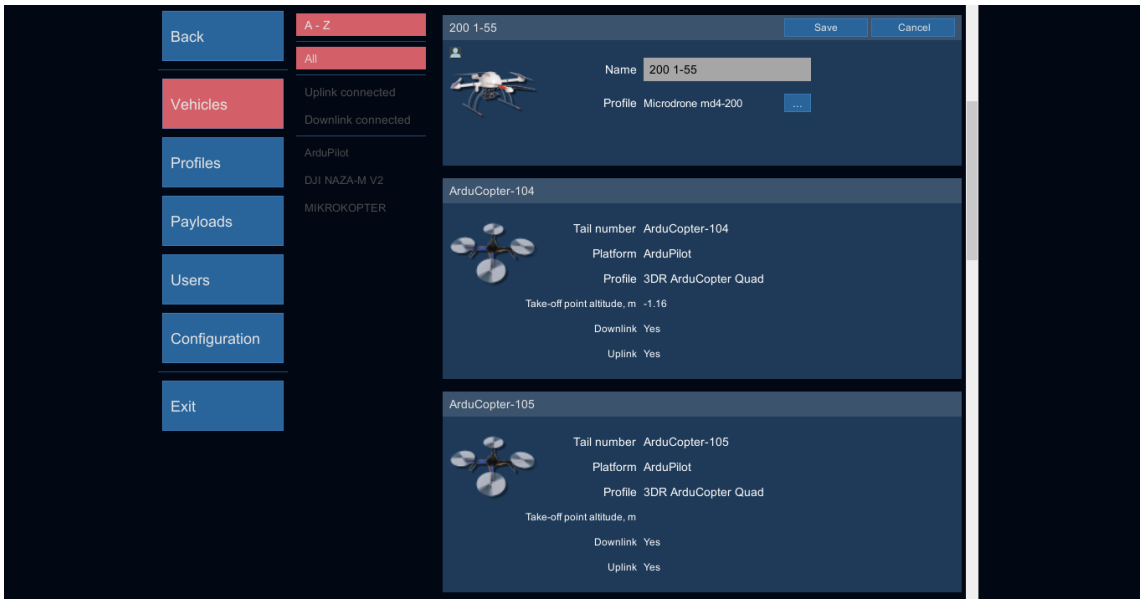

Figure 1: New Microdrones vehicle

4. Repeat steps above for each your Microdrones vehicle.

Note that if both uplink and downlink are connected only uplink will be active because the Microdrones stops telemetry sending after reception of commands via uplink. The downlink can be activated either by issuing "A-RM" command or by disconnecting uplink and re-inserting a battery.

1.2 Mission execution specifics

Please connect You Microdrones vehicle to computer before it reaches twin beep signal or "ready for take off" status to ensure correct mission upload.

1.3 Command execution specifics

Command	Support	Notes
ARM	Partial	The command issuing does not turn on motors but allows doing it from RC without battery re-inserting. It also activates telemetry sending.
DISARM	Partial	Does not stop motors but activates uplink channel and deactivates telemetry sending.
AUTOMODE	No	
MANUALMODE	No	
RETURNHOME	No	
TAKEOFF	No	

Copyright © 2015, Smart Projects Holdings Ltd

LAND	No	
EMERGENCYLAND	No	

1.4 Telemetry information specifics

Nothing specific.

1.5 Fail-safe actions

GPS Lost:

Action	Result
Wait	Aircraft tries to maintain altitude
Land	Aircraft slowly decends

RC Lost:

Action	Result
Wait	Aircraft returns home and lands
Land	Aircraft returns home and lands
Return Home	Aircraft changes altitude to failsafe ALT(50m) and returns home
Continue	Aircaft continues mission

Battery Low:

Action	Result
Wait	If possible aircraft returns home and lands, if not possible slowly decends
Land	If possible aircraft returns home and lands, if not possible slowly decends
Return Home	If possible aircraft returns home and lands, if not possible slowly decends
Continue	Aircaft continues mission

1.6 Configuration file

Default configuration file of the Microdrones VSM suits most needs and it is generally not necessary to modify it.

Configuration file location:

- **On Microsoft Windows:**

```
C:\Program Files (x86)\UgCS\bin\vsm-microdrones.conf
```

- **On GNU/Linux:**

```
/etc/opt/ugcs/vsm-microdrones.conf
```

- **On Apple OS X:**

```
/Users/[user name]/Library/Application Support/UGCS/configuration/vsm-microdrones.conf
```

1.6.1 Common parameters

All VSMs share a common set of configuration file parameters described in Common configuration file parameters. Microdrones VSM configuration file prefix is:

```
vehicle.microdrones
```

Copyright © 2015, Smart Projects Holdings Ltd

1.6.2 Serial port configuration

Mandatory. At least one serial port should be configured, otherwise VSM will not try to connect to the vehicle.

- **Name:** vehicle.microdrones.serial_port
- **Description:** Serial port configuration, for more details see Serial port configuration. Default Microdrones serial port communication speed is 38400 baud.
- **Example:**

```
vehicle.microdrones.serial_port.1.name = com1
vehicle.microdrones.serial_port.1.baud = 38400
```

1.6.3 Model name override

By default the VSM sets 'MD' model name for the Microdrones vehicles. It can be overriden to more specific name in the VSM configuration.

Example:

```
vehicle.microdrones.custom.my_drone.serial_number = 1102
vehicle.microdrones.custom.my_drone.model_name = MD4-200
```

In this example the model name for the drone with serial number '1102' is overriden to value 'MD4-200'.

1.7 Common configuration file parameters

VSM configuration file is a text file specified via command line argument - *-config* of the VSM application. Example:

```
--config /etc/opt/ugcs/vsm-ardupilot.conf
```

Each configuration parameter is defined as a line in the configuration file with the following structure:

```
name1.name2....nameX = value
```

where name1, name2 ... nameX are arbitrary names separated by dots to divide a variable into logical blocks and a value which can be a number value or a text string depending on the context. See below the description about common VSM configuration parameters.

1.7.1 UgCS server configuration

1.7.1.1 Listening address

Mandatory.

- **Name:** ucs.local_listening_address = [IP address]
- **Description:** Local TCP address to listen for incoming connections from UgCS server. Specify *0.0.0.0* to listen from all local addresses.
- **Example:** ucs.local_listening_address = 0.0.0.0

1.7.1.2 Listening port

Mandatory.

- **Name:** ucs.local_listening_port = [port number]
- **Description:** Local TCP port to listen for incoming connections from UgCS server. Default is 5556.
- **Example:** ucs.local_listening_port = 5556

Copyright © 2015, Smart Projects Holdings Ltd

1.7.2 Logging configuration

1.7.2.1 Level

Optional.

- **Name:** log.level = [error|warning|info|debug]
- **Description:** Logging level.
- **Default:** info
- **Example:** log.level = debug

1.7.2.2 File path

Optional.

- **Name:** log.file_path = [path to a file]
- **Description:** Absolute or relative (to the current directory) path to a logging file. Logging is disabled if logging file is not defined. File should be writable. Backslash should be escaped with a backslash.
- **Example:** log.file = /var/opt/ugcs/log/vsm-ardupilot/vsm-ardupilot.log
- **Example:** log.file = C:\\Users\\John\\AppData\\Local\\UGCS\\logs\\vsm-ardupilot\\vsm-ardupilot.log

1.7.2.3 Maximum single file size

Optional.

- **Name:** log.single_max_size = [size]
- **Description:** Maximum size of a single log file. When maximum size is exceeded, existing file is renamed by adding a time stamp and logging is continued into the empty file. [size] should be defined as a number postfixed by a case insensitive multiplier:
  - Gb, G, Gbyte, Gbytes: for Giga-bytes
  - Mb, M, Mbyte, Mbytes: for Mega-bytes
  - Kb, K, Kbyte, Kbytes: for Kilo-bytes
  - no postfix: for bytes
- **Default:** 100 Mb
- **Example:** log.single_max_size = 500 Mb

1.7.3 Serial port configuration

Optional. VSM which communicates with vehicles via serial ports should define at least one serial port, otherwise VSM will not try to connect to the vehicles. Port name and baud rate should be both defined. [prefix] is unique for each VSM.

1.7.3.1 Port name

Optional.

- **Name:** [prefix].[port index].name = [regular expression]
- **Description:** Ports which should be used to connect to the vehicles by given VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Expression is case insensitive on Windows. [port index] is a arbitrary port indexing name.
- **Example:** vehicle.ardupilot.serial_port.1.name = /dev/ttyUSB[0-9]+|com[0-9]+
- **Example:** vehicle.ardupilot.serial_port.2.name = com42

Copyright © 2015, Smart Projects Holdings Ltd

1.7.3.2 Port baud rate

Optional.

- **Name:** [prefix].[port index].baud.[baud index] = [baud]
- **Description:** Baud rate for port opening. [baud index] is an optional arbitrary name used when it is necessary to open the same serial port using multiple baud rates. [port index] is an arbitrary port indexing name.
- **Example:** vehicle.ardupilot.serial_port.1.baud.1 = 9600
- **Example:** vehicle.ardupilot.serial_port.1.baud.2 = 57600
- **Example:** vehicle.ardupilot.serial_port.2.baud = 38400

1.7.3.3 Excluded port name

Optional.

- **Name:** [prefix].exclude.[exclude index] = [regular expression]
- **Description:** Ports which should not be used for vehicle access by this VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Filter is case insensitive on Windows. [exclude index] is a arbitrary indexing name used when more than one exclude names are defined.
- **Example:** vehicle.ardupilot.serial_port.exclude.1 = /dev/ttyS.∗
- **Example:** vehicle.ardupilot.serial_port.exclude = com1

1.7.3.4 Serial port arbiter

Optional.

- **Name:** [prefix].use_serial_arbiter = [yes|no]
- **Description:** Enable (yes) or disable (no) serial port access arbitration between VSMs running on the same machine. It is recommended to have it enabled to avoid situation when multiple VSMs try to open the same port simultaneously.
- **Default:** yes
- **Example:** vehicle.ardupilot.serial_port.use_serial_arbiter = no

1.7.4 Network connection configuration

Optional. VSM which communicates with vehicles via network should define at least one network connection, otherwise VSM will not try to connect to vehicles. [prefix] is unique for each VSM.

1.7.4.1 Local IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_local_address = [IP-address]
- **Description:** Local IP-address to listen for incoming UDP packets on. Specify 0.0.0.0 if you want to listen on all local addresses.
- **Example:** vehicle.ardrone.detector.1.udp_local_address = 0.0.0.0

Copyright © 2015, Smart Projects Holdings Ltd

1.7.4.2 Local UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Example:** vehicle.ardrone.detector.1.udp_local_port = 14550

1.7.4.3 Remote IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_address = [IP-address]
- **Description:** Remote IP-address to send outgoing UDP packets to.
- **Example:** vehicle.ardrone.detector.1.udp_address = 192.168.1.1

1.7.4.4 Remote UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_port = [port number]
- **Description:** Remote UDP port to send outgoing packets to.
- **Example:** vehicle.ardrone.detector.1.udp_port = 14551

1.7.5 Mission dump path

Optional.

- **Name:** [prefix].mission_dump_path = [path to a file]
- **Description:** File to dump all generated missions to. Timestamp is appended to the name. Delete the entry to disable mission dumping. All directories in the path to a file should be already created.
- **Example:** vehicle.ardupilot.mission_dump_path = C:\\tmp\\ardupilot_dump

2 Disclaimer

DISCLAIMER OF WARRANTIES AND LIMITATIONS ON LIABILITY.

(a) SMART PROJECTS HOLDINGS LTD MAKE NO REPRESENTATIONS OR WARRANTIES REGARDING T-HE ACCURACY OR COMPLETENESS OF ANY CONTENT OR FUNCTIONALITY OF THE PRODUCT AND ITS DOCUMENTATION.

(b) SMART PROJECTS HOLDINGS LTD DISCLAIM ALL WARRANTIES IN CONNECTION WITH THE PRODUCT, AND WILL NOT BE LIABLE FOR ANY DAMAGE OR LOSS RESULTING FROM YOUR USE OF THE PRODUCT. INCLUDING BUT NOT LIMITED TO INJURY OR DEATH OF USER OR ANY THIRD PERSONS OR DAMAGE TO PROPERTY.

(c) THE SOFTWARE IS SUPPLIED AS IS WITH NO WARRANTIES AND CAN BE USED ONLY AT USERS OWN RISK.

Copyright © 2015, Smart Projects Holdings Ltd